DRAGON SYSTEMS' AUTOMATIC TRANSCRIPTION OF NEW TDT CORPUS

Larry Gillick, Yoshiko Ito, Linda Manganaro, Michael Newman, Francesco Scattone, Steven Wegmann, Jon Yamron, Puming Zhan

Dragon Systems, Inc.
320 Nevada Street, Newton, MA 02160

ABSTRACT

Dragon Systems has agreed to provide automatically generated transcripts for around 1000 hours of Broadcast News, annotated with word-level time-markings and confidence estimates. Pilot transcripts of about 30 hours of data will be available sometime in February, with the project completed by mid-July.

In this paper, we describe how we took our 1997 Hub4 evaluation system which ran at around 140 times real time, and modified it to run around 6 times faster with virtually no increase in error rate. We describe in detail the confidence algorithm, and show how it is a good predictor of which words were recognized correctly.

1. INTRODUCTION

As a service to the research community, Dragon Systems has agreed to provide automatically generated transcripts for around 1000 hours of Broadcast News recordings, expected to include programs from Voice of America, CNN Headline News, and ABC World News Tonight. The data will be of particular interest to the Topic Detection and Tracking community, and the annotated transcripts will be made available through the LDC. In addition to the raw transcripts, we will provide word-level time markings and confidence estimates.

The timetable for the project is as follows. By early February, we plan to have finished a pilot project to recognize 30 hours of data. The annotated transcripts will be distributed to sites to give them an early feel for what the data will look like, and how their algorithms will perform on errorful recognition transcripts. By the beginning of March, we will have frozen our system and be ready to recognize the (approximately) 1000 hours of data, which corresponds to 6 months of daily broadcasts from January through June 1998. Barring any delays with the acoustic data, the entire project should be finished by mid-July.

In the next section of this paper, we describe how we sped up our system to enable us to process such a large quantity of data, and in the final section we describe the algorithm for estimating word-level confidences.

2. SPEEDING UP THE SYSTEM

The 1997 Hub4 evaluation system ran at about 140 times real time (140xRT) on a 233 MHz Pentium-Pro with 256MB of memory. (For full details of the system, see the paper [1] elsewhere in this proceedings.) It would be prohibitively expensive to run at this speed on a large corpus, so we looked carefully at how to reduce the time required. Through careful tightening of recognition thresholds, and changing the structure of the language model, we were able to cut the time by a factor of 6 to around 25xRT with virtually no loss in recognition accuracy.

The pre-processing of the speech data originally took around 15xRT, employing simple acoustic and language models for several passes of fast recognition for use in speech detection, channel normalization, and speaker warping. Simply by tightening the pruning thresholds, we reduced this to around 4xRT, but we know we can still make significant improvements in speed in this part of the system.

The bulk of the required time is spent in two passes of slow, high beam-width recognition. The recognition transcript from the first pass is used for unsupervised adaptation through linear regression, to give adapted models which are used in the second pass. In total, the recognition takes around 125xRT. We tried to cut this time in three ways:

1) *Acoustic models*. We had anticipated a significant speed-up resulting from careful design of the models, varying both the number of output distributions, and the number of gaussian components in each distribution. In practice, we found that we could not speed up the recognition without increasing the error rate more than we were prepared to accept. The sad fact of life is that *big* models really do have lower error rates.

2) *Pruning thresholds*. By careful selection of pruning thresholds, we reduced the time from 125xRT to about 35xRT, with no measurable degradation in performance on the evaluation corpus. However, the tuning process was rather labor-intensive, and there is always the danger that the tighter thresholds will not work well with some new noisy (or otherwise mismatched) data.

3) *Language models*. The evaluation was run with a 3-way interpolated trigram language model, each component being trained from a different corpus. Such a language model is slow

for several reasons. First, we have to do a separate look-up of each trigram in each of the three language models. Second, once we have the scores, we have to interpolate them, a relatively inefficient process. Third, the sheer size of the language model results in inefficient use of memory and even swapping into virtual memory.

The alternative is a merged language model, where we weight the *n*-gram counts from the different corpora appropriately, to compensate for the radically different sizes of the corpora (which vary between 1 and 350 million words), and then build a single language model from this combined (fictitious) corpus. In this way, we achieved a speed-up of about one third, from 35xRT to 22xRT.

The downside of this approach is that it can be hard to find the optimal merging weights. For an interpolated language model, we only need to build each component model once, and then we can optimize the weights by running multiple recognition or perplexity tests at different settings. In contrast, for each set of merging weights, we need to process several gigabytes of data to build a fresh language model, a much slower and more cumbersome process. As a result, we cannot perform as exhaustive a search of parameter space, and indeed the merged language model performed about 0.5 points worse. Even if we found the best weights, the back-off and *n*-gram smoothing procedures will give different results for the two systems. In any case we cannot be sure whether the merged language model was intrinsically worse, or whether we were using sub-optimal weights.

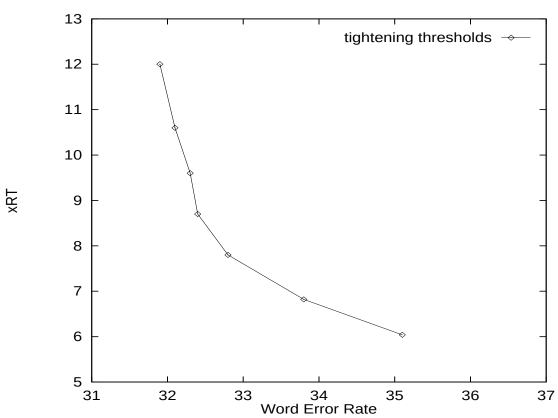

Figure 1: Tuning for broadcast news

To speed recognition further, we need to do a careful comparison of the trade-offs resulting from smaller models and/or tighter thresholds. As shown in the accompanying graph, we can get a further speed-up of almost 40%, at the cost of one percent absolute in recognition accuracy, by keeping the models fixed and turning down the thresholds. At that point, it is more efficient to keep the thresholds fixed and move to smaller models. (Numbers shown are for a small language model on an internal development test set.)

3. CONFIDENCE ESTIMATION

In addition to providing time-marked word-level output, the system also provides a confidence estimate for each recognized word. By "confidence" we mean the probability that the recognized word is correct. We have described our approach to confidence estimation and evaluation elsewhere (see for example [2]). To review: During the recognition pass, the recognizer computes values of various "predictors" which may provide insight into the accuracy of the transcription. These predictors are generated for each word of the recognition hypothesis and are combined via an appropriate statistical model to compute a confidence value. The Broadcast News transcription system uses logistic regression to combine the predictors, i.e. the confidence p that a word is correct is modeled by the formula

$$\log(\, p / (1 - p)\,) = a_0 + a_1 x_1 + \ldots + a_n x_n$$

where the x_i's are the predictor values and the a_i's are regression coefficients trained by optimizing performance on development data.

For the current system, we use six predictors:

- WDUR – the duration of the word.
- LM – the language model score for the word.
- NBEST – the fraction of times the word appears in the appropriate location in a list of the top N hypotheses for the utterance (we use $N = 100$ in the current system).
- ACTV – the average over the frames of the word of the number of HMM states (across the whole vocabulary) active in that frame.
- SCR – a normalized acoustic score for the word, defined as follows. For each frame of the word, we take the acoustic score of the state assigned to that frame, and we subtract the best possible acoustic score when scored against any of the active states in the system. Finally, this score difference is averaged over all frames of the word.
- ULEN – the log of the number of recognized words in the utterance.

Of course the framework is quite general: new predictors can easily be incorporated into the model, and alternate models for their combination, such as generalized additive models, can easily be substituted for the logistic regression.

During training, the recognizer dumps the values of the predictors for each word along with a label of whether or not the word is correct. This information is then used to train the logistic regression coefficients by maximizing the likelihood of the data. At run-time, the system loads the coefficients, computes the predictor values, and outputs a confidence estimate for each recognized word.

The likelihood we wish to maximize is defined as follows. For each word w_i in a recognized word string $w_1\ w_2 \ldots w_n$, we have an associated confidence p_i that the word is correct, and $(1 - p_i)$ that the word is incorrect. Thus the average log likelihood of observing the string of correct and incorrect labels associated to the word string $w_1\ w_2 \ldots w_n$ is given by

(1) $L = 1/n\ [\ \sum_{correct} \log(p_i) + \sum_{incorrect} \log(1 - p_i)\ ]$.

Of course, one could always output the overall correctness rate p for the confidence values p_i. Indeed a good way of assessing the performance of the confidence model is to see how much the likelihood defined in equation (1) improves on the value derived by using this uniform value of p.

	Broadcast News		Switchboard	
	value	*t-val*	*value*	*t-val*
intercept	1.17	9.4	1.54	9.9
WDUR	0.0117	9.2	4.02e-04	0.2
LM	-0.00629	-19.9	-0.00733	-13.3
NBEST	2.24	29.3	2.62	23.1
ACTV	-1.2e-05	-11.9	-1.01e-05	-6.8
SCR	-0.347	-41.1	-0.183	-17.9
ULEN	0.197	5.5	-0.219	-7.2

Table 1: Estimated parameters of confidence models for Broadcast News and Switchboard.

The confidence model used for the Broadcast News transcription effort was trained on the 1996 Hub4 development test (totaling about 19,000 words). The coefficients for the six predictors (as well as the intercept term) are shown in Table 1. The associated *t*-values are a measure of the relative value of the predictors. For comparison, the table also shows an analogous model trained for the Switchboard corpus of conversational telephone speech [3], a corpus which has been the subject of numerous studies in confidence estimation (for example, see [4] - [6] in addition to [2]). The Switchboard model was trained from 20 Switchboard conversation halves (totaling only about 9000 words).

Note the differences in the relative value of predictors for the two corpora. In particular, while Broadcast News places greatest importance on the normalized acoustic score, Switchboard favors the NBEST measure. The increased importance of the normalized acoustic score may be due in part to the considerable variability in channel types in the Broadcast News data versus the more uniform channel for Switchboard conversations. Also, utterances in Broadcast News tend to be longer than in Switchboard with the result that the *N*-best list concentrates its variability in the most confusable region of the utterance, making NBEST a less valuable predictor for many words. Also note that the sign of ULEN differs: i.e. longer utterances are easier to recognize in Broadcast News and harder for Switchboard. This may be because the longer Broadcast News utterances – such as news anchor monologues – tend to be in regions of cleaner speech. Some of these corpus-specific issues are examined in more detail below.

	Broadcast News		Switchboard	
	train	*test*	*train*	*test*
$\exp(L)$	0.644	0.645	0.597	0.602
$\exp(L_{base})$	0.569	0.570	0.538	0.533
p_{base}	0.748	0.749	0.689	0.670
#words	18960	22024	8969	18103

Table 2: Performance of confidence models on training and test data.

How well do these models perform? The models defined above were used to estimate confidences on the 1996 Hub4 evaluation test for Broadcast News and a 20-conversation test set for Switchboard. Table 2 gives the resulting value of $\exp(L)$ for these models. We find this exponentiated form of equation 1 more intuitive; it corresponds to the geometric mean of the predicted probabilities. The table also provides the values of p_{base} – the overall correctness rate for the transcription – and $\exp(L_{base})$ obtained by using this uniform value in place of the word-specific confidences in equation 1. [Note that the correctness rate p_{base} is somewhat higher than (1 - word error rate) since no account is taken of deletion errors – only performance on recognized words is tallied.]

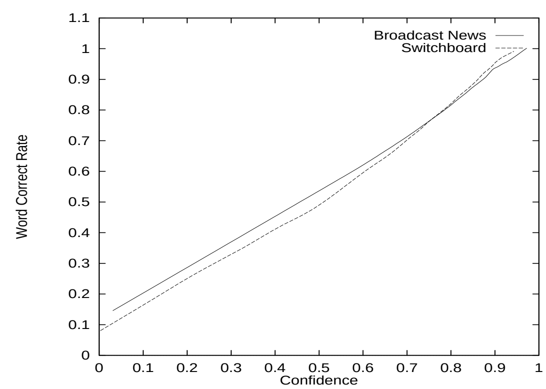

Figure 2: Word correctness rate vs. confidence estimate

A more graphical way of viewing performance is provided in Figures 2 and 3. The first of these shows the correctness rate for the transcribed words as a function of confidence value, computed over a moving window. As you can see, the confidence estimate is a good predictor of the true probability of correctness. Figure 3 gives the cumulative correctness rate for the top *x*% of the recognized words when ranked in terms of confidence score. For the Broadcast News task, if we limit attention, say, to the 40% of the recognized words in which we are most confident, the correctness rate is an impressive 95%. This bodes well for topic detection and tracking applications where speech recognition serves as a front-end.

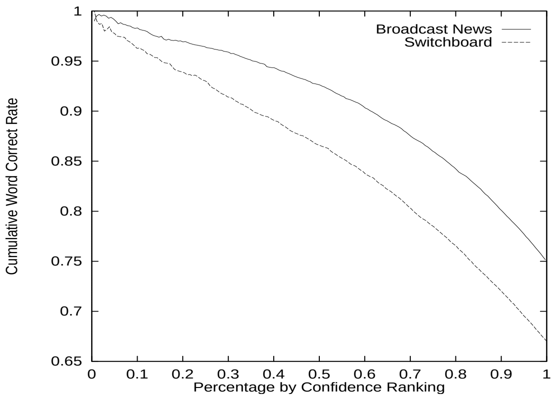

Figure 3: Cumulative word correctness rate when words are rank-ordered by confidence

Because Broadcast News data has been classified into a number of "conditions", we also looked at differences between prediction models customized for each channel type. To do this, we partitioned the development data using the F0, ..., FX labels and trained separate confidence models for each condition. The resulting coefficients are given in Table 3.

There are pronounced differences between the sets: for example, in the cleanest speech (F0) the language model score is one of the best predictors, while it is a relatively weak predictor in music (F3). We tried using these channel-specific models in a cheating experiment, using the known labels for the test set, but found very little average improvement. Perhaps this is because the training data is too badly fragmented to train good per-channel models, or the composite model could already be doing a more than adequate job.

4. CONCLUSIONS

We have demonstrated the ability to recognize broadcast news data at a speed which makes feasible the automatic, accurate transcription of a large corpus. Furthermore, we are greatly encouraged by the ability of the confidence model to select a subset of the words in the recognition transcript with a very low error rate. The community will surely find many uses for this technology, still only in its infancy.

5. REFERENCES

1. S. Wegmann et al, "Dragon Systems' 1997 Broadcast News Transcription System", in this proceedings.
2. L. Gillick, Y. Ito, and J. Young, "A Probabilistic Approach to Confidence Estimation and Evaluation," *Proc. ICASSP-97*, Munich, April 1997.
3. J. Godfrey et al., "SWITCHBOARD: Telephone Speech Corpus for Research and Development," Proc. ICASSP-92, San Francisco, March 1992.
4. E. Eide et al., "Understanding and Improving Speech Recognition Performance Through the Use of Diagnostic Tools," *Proc. ICASSP-95*, Detroit, May 1995.
5. S. Cox and R. Rose, "Confidence Measures for the Switchboard Database," *Proc. ICASSP-96*, Atlanta, May 1996.
6. M. Weintraub et al., "Neural-Network Based Measures of Confidence for Word Recognition," *Proc. ICASSP-97*, Munich, April 1997.

	F0 Clean, read	F1 Spontaneous	F2 Narrow band	F3 Bkg music	F4 Noise	F5 Non-native	FX Other
intercept	0.905 [2.2]	1.79 [6.8]	1.26 [3.7]	0.410 [1.0]	1.31 [3.1]	1.16 [2.0]	1.27 [4.7]
WDUR ($x10^{-3}$)	26.6 [6.4]	5.43 [2.0]	19.6 [6.0]	0.298 [0.1]	22.7 [5.4]	27.1 [4.5]	8.53 [3.0]
LM ($x10^{-3}$)	-10.7 [-11.5]	-5.1 [-7.8]	-6.58 [-8.1]	-3.57 [-3.7]	-7.99 [-8.0]	-9.05 [-6.6]	-7.03 [-9.5]
NBEST	3.24 [15.1]	2.04 [12.4]	2.13 [11.2]	1.93 [7.5]	2.97 [13.3]	2.87 [10.0]	1.71 [9.7]
ACTV ($x10^{-6}$)	-14 [-2.3]	-2.1 [-1.1]	-5.98 [-1.7]	-2.27 [-0.8]	-12.8 [-3.4]	-7.14 [-1.1]	-10 [-6.6]
SCR ($x10^{-3}$)	-288 [11.1]	-379 [-23.1]	-3150 [-14.6]	-401 [-12.8]	-336 [-12.2]	-291 [-9.0]	-257 [-13.3]
ULEN ($x10^{-3}$)	240 [1.9]	6.17 [0.1]	-129 [-1.1]	601 [4.9]	-56.3 [-0.4]	114 [0.6]	22.7 [0.3]

Table 3: Model parameters when confidence models are separately trained for each channel condition. The top number gives the value of the coefficient and the bracketed number the associated *t*-value.